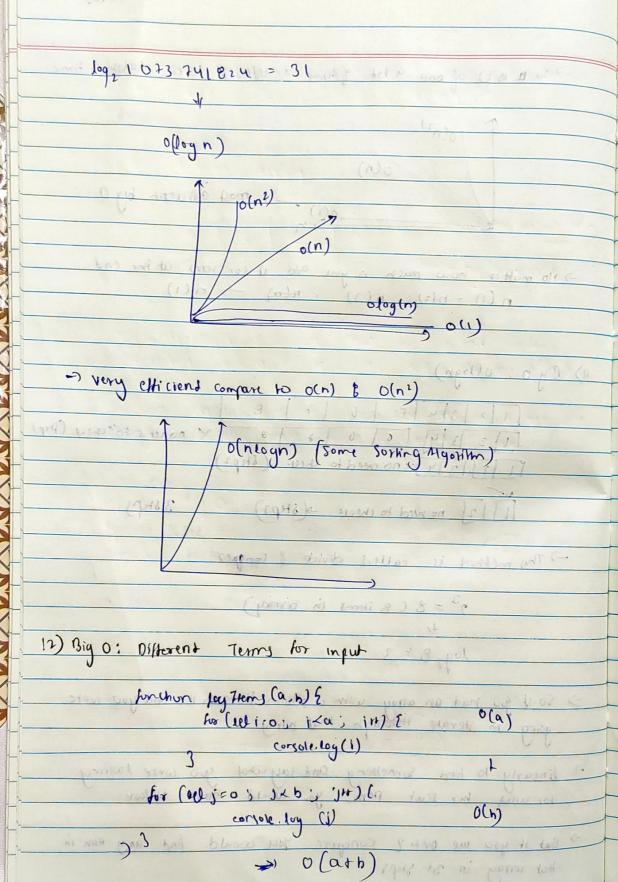
9) Bigo: Prop non-Dominants	pal on Days	
	(1-10)	
function lag Iterri(n) [for(letieo; i <n; i+t)="" td="" {<=""><td>vol.</td></n;>	vol.	
for (letiens ix n ; itt) {		
dor (jej=0 s j <n; s<="" td=""><td>14+) { (n)2</td></n;>	14+) { (n)2	
6 - 10 1 6	.: ()	
Consoit. Jeg (
3 (1) wet 3/5(no)		
for(1et 1 =0; 1 < n; k+1){	(n)	
consule. Log (1<)	į į	
3	(a) mittet	
Sacio + our + our	· ·	
dog Thims (100)	0 (n2+n)	
- Complete date		
-> If we book n = 100, men Square would	be 10,000 where	
-> If we book n = 100, then square would the single and that added to it this e	equation of only 100	
-> gt's not seally affecting me no up operation	wn,	
gues nucleur a print our print the sol o	Dus vil 140 02 (+	
M Squared & the dominant term and n by	Itself to the hon-	
dominant term, so just remove it.		
	(Sta)0 0 pd (9	
-> ove drop non dominance.	· ·	
(x)20111		
12 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	Ind:	
10) Big O: O(1) : constant time.		
fundin add Tem (n) f		
return non -> n.		
	2 Page	
1((°(1)0)	(4)	
	00	

-> 50 a to 0 of one a lot of time is reflected to an constant time
10(n2)
och (a gelja
o(n)
most efficient by O.
o.(1) n most efficient by 0.
2 at a signal of the god
-> No matter how much in you add it constains at the end
$n(1) - n(2) \dots n(3) \dots n(n) \rightarrow n(1)$
(00
11) Big o: o(logn)
(in) of too of money board (in)
[1/2/3/4/5/6/7/8]
1 2 3 4 6 6 7 8 X nonced to check (step)
1 2 3 4 × no need to will (Step2)
[2] no need to check - (Steps) 3 steps
18 19 Wall to Order Torder
—) The state of th
-) This method is called divide & conquer.
2
$2^3 = 8 (8 \text{ Stems in array})$
4
log 28 = 3 by 10 compl +0x compl : 0 10 (C)
> so it you had an array wirm a billion Items in it, and you were
> 50 It you had an array wim a billion Items in it, and you were going to sterate through that array
(1) 401 14 14 14
-) linearly to test 5 method and every that the tiere two
or way he lost them, you awald have to lower
for way the last them, you arrang many to make
the put styrios
-> But it you we prite & conquire you would had any Hom in
that array in 31 steps ()



> 91 it is neated loop
(a) i but was a ribbon do be
function dog Herns (a, b) {
ty (let i=0; i(a; i++) (0(a)
bo (105-0, 526; JH) (0(b)
3 3
1) autor pa some
♦ O(a+b)
(A) O XXXII VA ADADI CO S I CO
13) Bly O: Arrays.
0(1)
11] 3 22 7 my tray, pun(11) -7 add me 1km of end
0 1 2 3
my (may, pop () - remove the item
how may because your gency to reliase a it again
-) from born me operation are did not
seastange the indexel. 80 it is O(1)
push and pop -> O(1)
462.00
A My Mray. Shift() -) remove from from:] O(n) unshift(1) -) add the element in front
· unshift(1) - add the element in front
3 million (1938)
(post)
=) My Amay. Splice (1, 0, 'Hir)
1 (political)
(sayone & any Index no need to
basimen to sewant lugar
add 'Mi' ikm
(states)
11 'ni" 3 23 221
J, you in
0(n)

-) So It doesn't matter it you'r remove (or) adding Somewhere		
-) So it doesn't matter it you'r remove (or) adding Somewhere in the middle of array that is o(n) me		
1 C A LOUIL DE CONTRA		
(4)0)(4) (4) (4)		
-) find me clement away		
search by value o(n)		
Search by value o(h)		
U 1 2) -> Search by index O(h)		
(2) Ano : drawns		
-) The big advantages of array & that we can find something		
in an array with a million thems that is always O(n)		
But disadvantages of array it you want odd in the big begin		
ming, because yours going to reinder it again.		
that his are growings and med med to		
(DO 11 11 08 NAMED IN SPACE DOS		
(4) Wrap up: Big O.		
(,000,000		
10 of object = 10,000.		
1 OCN		
Tobb minum		
1000 million a 1000.000		
(Oln) =0100),100000000000000000000000000000000		
(Proportional)		
(Divido & conquer)		
30(logn) ≈ 2 ≈ 10		
(1) = 1, 1 no. of opposition		
(Constant)		
N=100 EE E 110 001=1		
n= 1000		
(1)0		

-> lbig	Ocheatikeet con Denible
	o(n) > fur
	good,

-> Sood string or different hind kind of dale other men no.

Me best time complexity you can get hird farring

algorithm is O(n)